

Claims:

What is claimed is:

1. A system for determining dependencies between J2EE components, comprising:
 - a first application server that includes an application deployed thereon; and
 - a preprocessor component that can be used to interrogate the applications' deployment information and any dependencies included therein, and communicate a subset of that information to a second application server for use in deploying the application at the second application server.
2. The system of claim 1 wherein said preprocessor performs the steps of:
 - interrogating the deployed application at the first application server to find all JNDI names present in the application;
 - determining which of said JNDI entities will be realized at runtime;
 - parsing through both an application-side list, and a server-side list, and locating dependencies that correlate with one another; and,
 - communicating application configuration information for use in deploying the application on said second application server.
3. The system of claim 1 wherein the configuration information is saved to a configuration file for subsequent use in deployment.
4. The system of claim 1 wherein the system further comprises a graphical user interface or web interface that allows the developer to select an application at a first server for subsequent deployment at a second server.

5. The system of claim 1 wherein the application side defines any EJBs used in the application and the resources dependent thereon., and the server side defines management APIs used by the application, data sources, and JMS queues.
6. A method for determining dependencies between J2EE components, comprising the steps of:
 - providing a first application server that includes an application deployed thereon; and,
 - interrogating the applications' deployment information and any dependencies included therein, and communicating a subset of that information to a second application server for use in deploying the application at the second application server.
7. The method of claim 6 wherein said preprocessor performs the steps of:
 - interrogating the deployed application at the first application server to find all JNDI names present in the application;
 - determining which of said JNDI entities will be realized at runtime;
 - parsing through both an application-side list, and a server-side list, and locating dependencies that correlate with one another; and,
 - communicating application configuration information for use in deploying the application on said second application server.
8. The method of claim 6 wherein the configuration information is saved to a configuration file for subsequent use in deployment.
9. The method of claim 6 wherein the system further comprises a graphical user interface or web interface that allows the developer to select an application at a first server for subsequent deployment at a second server.

10. The method of claim 6 wherein the application side defines any EJBs used in the application and the resources dependent thereon., and the server side defines management APIs used by the application, data sources, and JMS queues.

11. A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:

providing a first application server that includes an application deployed thereon; and,

interrogating the applications' deployment information and any dependencies included therein, and communicating a subset of that information to a second application server for use in deploying the application at the second application server.

12. The computer readable medium of claim 11 wherein said preprocessor performs the steps of:

interrogating the deployed application at the first application server to find all JNDI names present in the application;

determining which of said JNDI entities will be realized at runtime;

parsing through both an application-side list, and a server-side list, and locating dependencies that correlate with one another; and,

communicating application configuration information for use in deploying the application on said second application server.

13. The computer readable medium of claim 11 wherein the configuration information is saved to a configuration file for subsequent use in deployment.

14. The computer readable medium of claim 11 wherein the system further comprises a graphical user interface or web interface that allows the developer to

select an application at a first server for subsequent deployment at a second server.

15. The computer readable medium of claim 11 wherein the application side defines any EJBs used in the application and the resources dependent thereon., and the server side defines management APIs used by the application, data sources, and JMS queues.

16. A system for readily deploying applications from a first server to a second server, comprising:

a first server having an application deployed thereon;

a second server adapted to receive said application;

a preprocessor that interrogates the application's deployment information as deployed on said first server, and any dependencies included therein, and generates or communicates a subset of that information, for use in deploying the application at said second server.

17. The system of claim 16 wherein said preprocessor performs the steps of:
interrogating the deployed application at the first application server to find all JNDI names present in the application;

determining which of said JNDI entities will be realized at runtime;

parsing through both an application-side list, and a server-side list, and locating dependencies that correlate with one another; and,

communicating application configuration information for use in deploying the application on said second application server.

18. The system of claim 16 wherein the configuration information is saved to a configuration file for subsequent use in deployment.

19. The system of claim 16 wherein the system further comprises a graphical user interface or web interface that allows the developer to select an application at a first server for subsequent deployment at a second server.

20. The system of claim 16 wherein the application side defines any EJBs used in the application and the resources dependent thereon., and the server side defines management APIs used by the application, data sources, and JMS queues.

21. A method for readily deploying applications from a first server to a second server, comprising the steps of:

providing a first server having an application deployed thereon;

providing a second server adapted to receive said application;

interrogating the application's deployment information as deployed on said first server, and any dependencies included therein, and generating or communicating a subset of that information, for use in deploying the application at said second server.

22. The method of claim 21 wherein said preprocessor performs the steps of:
interrogating the deployed application at the first application server to find all JNDI names present in the application;

determining which of said JNDI entities will be realized at runtime;

parsing through both an application-side list, and a server-side list, and locating dependencies that correlate with one another; and,

communicating application configuration information for use in deploying the application on said second application server.

23. The method of claim 21 wherein the configuration information is saved to a configuration file for subsequent use in deployment.

24. The method of claim 21 wherein the system further comprises a graphical user interface or web interface that allows the developer to select an application at a first server for subsequent deployment at a second server.

25. The method of claim 21 wherein the application side defines any EJBs used in the application and the resources dependent thereon., and the server side defines management APIs used by the application, data sources, and JMS queues.

26. A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:

- providing a first server having an application deployed thereon;

- providing a second server adapted to receive said application;

- interrogating the application's deployment information as deployed on said first server, and any dependencies included therein, and generating or communicating a subset of that information, for use in deploying the application at said second server.

27. The computer readable medium of claim 26 wherein said preprocessor performs the steps of:

- interrogating the deployed application at the first application server to find all JNDI names present in the application;

- determining which of said JNDI entities will be realized at runtime;

- parsing through both an application-side list, and a server-side list, and locating dependencies that correlate with one another; and,

- communicating application configuration information for use in deploying the application on said second application server.

28. The computer readable medium of claim 26 wherein the configuration

information is saved to a configuration file for subsequent use in deployment.

29. The computer readable medium of claim 26 wherein the system further comprises a graphical user interface or web interface that allows the developer to select an application at a first server for subsequent deployment at a second server.

30. The computer readable medium of claim 26 wherein the application side defines any EJBs used in the application and the resources dependent thereon., and the server side defines management APIs used by the application, data sources, and JMS queues.